AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method for transitioning ~~altering~~ a network routing function in a network ~~with flow control on the link level without dropping data packets, said altering of the network routing is the transition~~ from a first routing function R_{old}, defining an established set of possible ~~connection~~ connections for forwarding data packets between a plurality of communication input ports $I_1,..,I_n$ and output ports $O_1,..,O_m$ ~~in a~~ of each network element in said network, to a second routing function R_{new}, defining a new set of possible ~~connection~~ connections between the said input and output ports of each network element, wherein the transitioning ~~is executed by the network element for transmitting and receiving data packets~~ of said network routing function ~~action in said network~~ is controlled ~~globally coordinated~~ by means of tokens defining said second routing function R_{new} to be used by each network element in the network to ensure that forwarding of data packets in the network elements in said network will not be halted indefinitely when altering the network routing function, where said method when applied to a network with link-level flow control will not create network deadlock, said method comprising:

(1) ~~for each input port I_i,~~ performing the following sequence of steps for each input port I_i of each network element in said network for altering the routing function used by each network element:

(1a) applying the first routing function R_{old} for input port I_i,

(1b) receiving a token on input port I_i,

(1c) stopping [[the]] forwarding of data packets [[from]] arriving on port I_i ~~arriving after said token~~,

(1d) applying the second routing function R_{new} for input port I_i,

(1e) starting forwarding of data packets to every output port O_j associated with said input port I_i according to the second routing function R_{new} only if said output port O_j has transmitted a token,

(2) ~~for each output port O_j,~~ performing the following sequence of steps for each output port O_j, of each network element in said network:

(2a) determining if the token has been received on all input ports I_i associated with the output port O_j according to the first routing function R_{old},

(2b) transmitting the token on the output port O_j when the token has been received on all said associated input ports I_i.

2. (Previously Presented) The method according to claim 1, wherein the network element is a switch.

3. (Previously Presented) The method according to claim 1 or 2, wherein the token is included in a data packet.

4. (Previously Presented) The method according to claim 1, wherein the method is applied to deterministic routing functions.

5. (Previously Presented) The method according to claim 1, wherein the method is applied to adaptive routing functions.

6. (Previously Presented) The method according to claim 1, wherein the method is applied to source routing.

7. (Previously Presented) The method according to claim 5, wherein if the adaptive method gives rise to a cyclic dependency graph, the graph is pruned into a non-cyclic one before the method is applied.

8. (Previously Presented) The method according to claim 1, wherein the method is applied to only parts of a complete network.

9. (Previously Presented) A network element, comprising

a plurality of output ports for transmitting data packets to other network elements in a network,

a plurality of input ports for receiving data packets from other network elements in the network,

a processing device,

a memory ,

characterized in that the processing device is arranged to perform a method according to claim 1.

10. (Previously Presented) The network element according to claim 9, wherein said routing functions are implemented as tables stored in said memory.

11. (Previously Presented) The network element according to one of the claims 9 or 10, wherein said memory comprises computer program instructions arranged to perform said method when executed by said processing device.

12. (Previously Presented) A computer network system, comprising a number of network elements according to claim 9.

13. (Previously Presented) A computer program, embodied on a storage medium or in a memory, for execution by a processing device in a network element, characterized in that the program comprises a set of instructions arranged to perform a method according to claim 1 when executed by the processing device in the network element.